2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

A new fuzzy visual servoing with application to robot manipulator

Marco A. Moreno-Armendariz, Wen Yu

Abstract— **Many stereo vision algorithms require the images which are very similar in appearance, and the distance between the two cameras should be small enough. Normal visual servoing of robot manipulator needs both image information and joint velocities. In this paper we make two modifications to overcome these problems: 1) a new simple stereo vision model is proposed, 2) a fuzzy controller based on only visual information is implemented to a robot manipulator. We successfully apply our new algorithms to real-time visual servoing of a industrial robot. The stereo model proposed in this paper can also be extended to other applications such as, mobile robots, 3D reconstruction, etc.**

I. INTRODUCTION

The basic idea of stereo computer vision is to reconstruct objects using the information of two well-selected sensors. These sensors are separate by a constant distance, which permits to know the depth of the object. In general, recognition of 3D object requires two or more appropriately defined 2D images. For human being, corresponding points in the visual cortex in the process are called binocular stereopsis. Years of research have shown that determining stereo correspondences by computer is not an easy problem [6]. Many methods have been proposed for stereo vision, such as structure analysis from motion [10][12], stereo correspondence calculation [9][6][4] and shape reconstruction [11]. These methods can be described as four-steps process: 1) use visual algorithms to find features in an image, 2) find the same features in the other camera, 3) study the differences of the same point between the two cameras, 4) calculate the actual position of the feature point by the parameters of the cameras.

For real applications, many methods are subject to one or more limitations. It is not easy to find a direct relation in the images of two cameras. A trade-off between efficiency and accuracy is often necessary, such that a satisfied accuracy can be achieved while large computation is avoided. Many current methods are successful when the images are very similar in appearance, as in the case of human vision. This requires the two cameras have to be put as close as possible, and the objects should also be close to the cameras. When the distance between the cameras becomes large, surfaces in the images exhibits different degrees of foreshortening, there are large disparities in their locations in the two images. All of these make it difficult for computers to determine correct stereo correspondences. In this paper we present a new visual algorithm, which is an extend of the famous Weak-Perspective camera model [13]. It can process the images quickly and efficiently, and can realize auto-calibration.

The use of a vision system to complete the feedback control for a robotic system is referred as visual servoing [7]. It has the potential to provide a low-cost and a low-maintenance automation solution for unstructured industries and environments. Visual servoing is a rapidly maturing approach for robot manipulator control that is based on the visual perception of a robot and a work piece location. It is well known that most of industrial manipulators are equipped with the simplest proportional and derivative (PD) controller. For camera based visual servoing, PD-like control is also very popular [8][3][5],

$$\tau = -K_p [x_s - x_s^*] - K_d \left[\dot{q} - \dot{q}^*\right]$$

where x_s and x_s^* are the positions of end-effector and target in the images, $\dot{q}$ and $\dot{q}^*$ are real and desired joint velocities, K_p and K_d are positive matrices. This PD controller requires measurements of both images and joints velocity. It is very important to realize visual servoing with only vision information. In this paper, we propose a novel fuzzy controller based on our stereo vision algorithm. Several experiments show excellent behavior of the stereo vision algorithm and the fuzzy control, even when a large amount of salt and pepper noise are presented. This research is based on the following assumptions: 1) The visual marks attached to the robot manipulator are completely visible. 2) The target point is static and it is inside the visual space. 3) There exists at least one joint position $q^* \in \Re^3$ (the robot is 6 DOF, 3 links are corresponded to rigid robot manipulator, the other 3 links are on the robot´s hand), such that the end-effector of the robot is able to reach the target point. $q^* \in \Re^3$. Assumption 1 limits the motion of the robot. This means that the first degree of freedom of the robot could be to rotate $\pm 30°$ (see Fig. 1). Assumption 2 means that the vision algorithm proposed in this paper is suitable for fixed points. Assumption 3 gives a possibility to locate only the position (without orientation) of the end-effector of the robot and to design a fuzzy visual servoing.

II. STEREO VISION PLATFORM AND VISION ALGORITHM

The fuzzy control system based on computer vision is shown in Fig.1. The main parts are: A Robot Manipulator, CRS Robotics A465. A pair of analog cameras, Pulnix TM72-EX. A pair of lenses, Cosmicar C815B . A frame grabber, NI-1408. A personal computer with all the algorithms presented in this paper. An articular controller, CRS Robotics C550C.

This work was supported by CONACyT under Grant 38505A.

Marco A. Moreno-Armendariz is with Escuela de Ingeníeria, Coordinación de Investigación , Universidad La Salle, Benjamin Franklin 47 Col.Condesa, Mexico D.F, 06140, Mexico.

Wen Yu is with the Departamento de Control Automatico, CINVESTAV-IPN, Av.IPN 2508, México D.F., 07360, México yuw@ctrl.cinvestav.mx

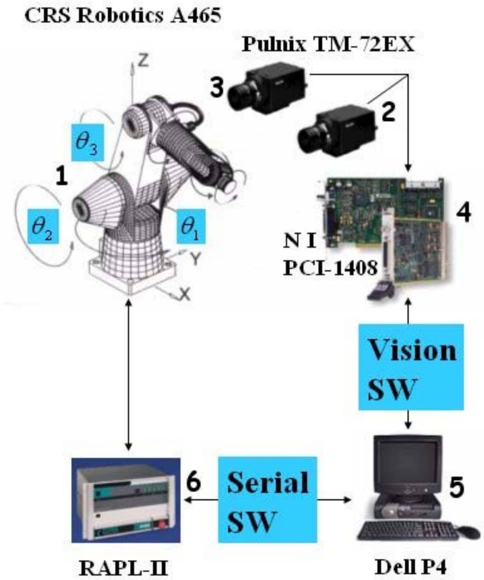
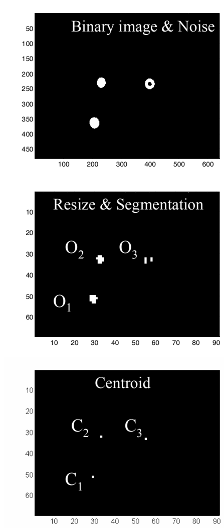


Fig. 1. Structure of Computer Vision System.

In order to reduce noise influence, the scene background is set to be black and only three white marks are put on the robot. These marks are attached to three main parts of the robot: base, elbow and end-effector. The mark of end-effector has a black circle inside it. We know the initial position (ready position) of the end-effector of the robot. The other two marks are used as references. The goal of vision algorithm is to detect the three white marks from images, and obtain the centroids of the marks such that we can use these values in optical stereo vision model which will be explained in the next section. We use following four steps to process the images: 1) Binaries the image. We establish an appropriate threshold value to binaries the image. The PCI-1408 device can automatically execute this process. 2) Apply a resize process to the binary image. Since the visual marks (circles) have a diameter of $d = 26$ pixels, we only need to search for objects which have this size. Any smaller objects will be considered as noise. However the whole size of the binary image is 640×480 pixels, there are too much information to be processed continuously. In order to work effectively, we resize the binary image by the following factor:

$$q = \frac{d \cdot \sin(45^\circ)}{2.5} \tag{1}$$

We just work with the resized image considerably reducing the amount of information to be computed. To obtain the resize value q, we made a benchmark test which can calculate execution time via *q-values*. q is changed from 1 to 25 ($q = 1$ means that no resize is applied, binary image is 640×480 pixels). We found that the best value is $q = 7$, because this value has a better balance between the execution time and the amount of pixels that represents the visual marks. So every object in the binary image will be represented in the resized image with at least 3 pixels. The final size of the image is 91×68 pixels. Make a segmentation of the visual marks which are obtained by the resize process. After resizing the image we have a list of pixels (its coordinates) that represents the three visual marks. By the mechanical behavior of the robot, we know that the visual marks of the robot never overlap. An example of our visual algorithms are presented in Fig.2. The first column of table presents the coordinates of the pixels obtained by the resize process. The second column shows the coordinates that are separated in three objects by the segmentation process. The third one is the centroid for the three visual points.4) Find the centroid of the visual marks. Now we present an algorithm to relate and label each centroid of the marks (note that one of the visual marks has a black circle inside). At ready position , the centroids are shown in Table 1.



Resize (x,y)		Segmentation (x,y)			Centroid (x,y)		
31	32	O_1	31	32	C_1	267.5	213
31	33		31	33			
32	31		32	31			
32	32		32	32			
32	33		32	33			
32	34		32	34			
32	55		33	32			
32	58		33	33			
33	32		33	34			
33	33		34	33			
33	34		34	32			
33	55	O_2	32	55	C_2	269	283
33	58		32	58			
34	32		33	55			
34	33		33	58			
34	55		34	55			
50	28	O_3	50	28	C_3	411	207.5
50	29		50	29			
50	30		50	30			
51	28		51	28			

Fig. 2. Example of appling the visual algorithms to obtain the centroid of the three white marks.

Table 1. LIST OF CENTROIDS TO BE LABELED

Object	*Centroid*	
	u	v
Object O_1	267.5	213
Object O_2	269	283
Object O_3	411	207.5

If there are black pixels inside the mark it is end-effector. The separation of the other two marks is made by comparison of v-values of them, the bigger one is labeled as elbow and the smaller one is base. For the Table 1, we label the marks as in Table 2.

Table 2. CENTROIDS RELATED WITH A PART OF THE ROBOT

Label	*Centroid*	
	u	v
End-effector	267.5	213
Elbow	269	283
Base	411	207.5

With above algorithm the computer can acquire unique correspondence between the visual mark and the robot's arm. A minimization requirement of the visual algorithm is the robot and its work space should appear complete in the images. We use a pair of lens, Cosmicar C815b. The focal length is $8.5mm$. A camera with this kind of lens at two meters to the scene can capture images in working space as $1194 \times 1728mm$. This distance satisfies the minimum requirement of the industrial robot we used. However, such big working space produces a distortion (named radial distortion). Optical stereo vision system is based on the centroid of the white marks. It's very important to eliminate the radial distortion. We use the algorithm presented in [13],

$$r = \frac{r'}{1 + d_2 r'^2 + d_4 r'^4} \tag{2}$$

where r is a corrected radius and r' is the distortion radius. In our experiment we obtain that $d_2 = -6.166e^{-7}$ and $d_4 = 2.083e^{-12}$. The coordinates without distortion are obtained by

$$\begin{aligned} x &= \frac{x' - X}{1 + d_2 r'^2 + d_4 r'^4} \\ y &= \frac{y' - Y}{1 + d_2 r'^2 + d_4 r'^4} \end{aligned} \tag{3}$$

where (x', y') are the coordinates of the centroid of the marks in the distortion image, (x, y) are the new coordinates without distortion, (X, Y) are the coordinates of the center of the image. This section solves the first part of the stereo correspondence problem [13], the second part of this problem is to determine the position of end-effector, which will be solved by the optical stereo vision model in the next section.

Now we discuss some concepts of stereo vision. An optical camera model is shown in Fig.3A relation between

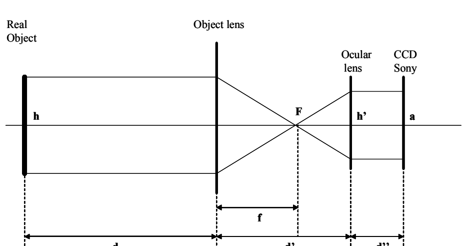


Fig. 3. Variables defined for the optical model.

the distance and height is

$$\frac{h'}{h} = \frac{f}{d' - f}, \quad \frac{1}{f} = \frac{1}{d'} - \frac{1}{d} \tag{4}$$

where d, d', d'', h, h' and f are defined in Fig.3, $d' > f$. In real application, f and d' are obtained by

$$f = \frac{dh}{h + h'd^2} \quad d' = \frac{h'd}{h} \tag{5}$$

Since the units of h and h' are centimeters and pixels, we have to use a factor as $k_f = 0.002708333$. With this factor we obtain $f = 0.585091$ and $d' = 1.704799$.

We first introduce a simple stereo model, the two cameras are mounted in the same plane with the distance of b_x. The position of the two cameras in world frame may be defined as $[-b_x/2, 0, 0]^T$ and $[b_x/2, 0, 0]^T$. Image planes are in front of each camera with a distance f. We assume that z-axis of the camera is the optical axis. Now the problem is how to use the image information (u_L, v_L) and (u_R, v_R) to the corresponding world frame (x, y, z). Using so-called pinhole lens model [6], when $f_R \neq f_L$ and the two camera axes are not parallel

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda_L Q_L \begin{bmatrix} u_L \\ v_L \\ f_L \end{bmatrix} + \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \tag{6}$$

Usually there not exit λ_L and λ_R such that $[x_1, y_1, z_1] = [x_2, y_2, z_2]$, but we may find (λ_L, λ_R) such that $\left\| [x_1, y_1, z_1]^T - [x_2, y_2, z_2]^T \right\|^2$ is minimized, i.e.,

$$\varepsilon^2 = \| \lambda_L P_L - \lambda_R P_R + t_L - t_R \| \tag{7}$$

Since the relation (7) between the world frame and the images of the two cameras is not an analytical expression. An approximation approach, named visual space method can be used [4],

$$\begin{aligned} x^v &= \frac{b_1}{\lambda \sin \psi} u_r \\ y^v &= \frac{b_1}{\lambda \sin \psi} v_r \\ z^v &= \frac{-b_1}{\lambda \sin \psi \tan \psi} (u_l - u_r) \end{aligned} \tag{8}$$

The error between $[x^v, y^v, z^v]$ and $[x, y, z]$ is defined as $[e_x, e_y, e_z]$. Now we discuss our new stereo vision model, it is another approximation and is simpler than [4]. This model is used for locating the end-effector of robot using visual information, it can calculate the 3D position of robot by the centroids of the object and the rotated angles of the cameras. The top view of the robot is show in Fig.4. Here c is the distance between the c_1 mark and the c_2 mark. The full geometric model is shown in Fig.5, perceive that the optical camera model is incorporated (the definitions of d, d' and d'' can be found in Fig.3). The position of the end-effector is obtained by the following procedure:

1) Calculate the angles θ_{1a} and θ_{1b} for the left and right cameras with respect to *x-axis* of the robot, see Fig.4). The values of these angles can be used while the cameras are fixed. The visual system is setup such that the *xz-plane* of the robot and the *uv-plane* of the camera are parallel when the **base** and the **elbow** visual marks are aligned, see Fig.4. To do it, we use the following steps: (a) Set the robot in Ready Position (see Fig.1). (b) Rotate the θ_1 angle *(joint 1)* of the robot by $0.25°$ (see Fig. 1) and capture an image

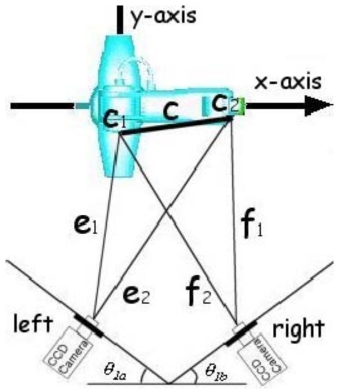


Fig. 4. Distances between the cameras and the robot.

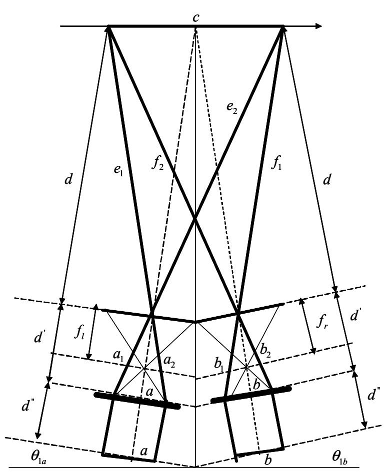


Fig. 5. Full geometric diagram used to obtain the distances between the cameras and the robot.

with the right camera. (c) Apply our visual algorithms to locate the centroid of the three visual marks and store it. (d) Compare the values of the v_l which coordinate the visual marks of base and elbow, if the values are the same (meaning aligned) go to step (e), if not return to (b). (e) Stop to move the robot and read the angle θ_1 (*joint 1*) from the controller. The value of this angle is stored as θ_{1b} and corresponds to the angle of the right camera. (f) Repeat the procedure using the left camera and store the value as θ_{1a}.

2) Select elbow and end-effector visual marks (c_1, c_2) and calculate the centroid coordinates (u_l, v_l) and (u_r, v_r) for both points in each image. Calculate the distance between the two points in the left and right images and set the distance as a and b respectively, see Fig.6.

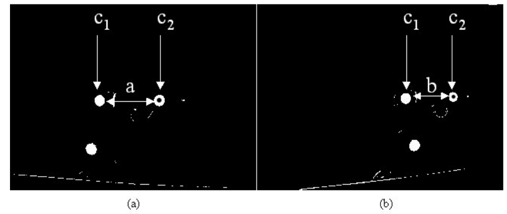


Fig. 6. Elbow visual mark (c_1) and end-effector mark (c_2) . The distance between these points: a left camera, b right camera.

3) From Fig.5 we obtain the next group of equations

$$\beta_1 = \begin{cases} \tan^{-1}\left(\frac{d'}{L_2 - L_1}\right), & L_1 < L_2 \\ 90^o + \tan^{-1}\left(\frac{L_1 - L_2}{d'}\right), & L_1 \geq L_2 \end{cases}$$
$$a_1 = \sqrt{(d')^2 + (L_2 - L_1)^2}$$
$$a_2 = \sqrt{(a_1)^2 + (a)^2 - (2\ a\ a_1 \cos(\beta_1))}$$
$$\beta_2 = sen^{-1}\left(\frac{a}{a_2} sen(\beta_1)\right), \beta_3 = 180° - \beta_1 - \beta_2$$
$$\beta_6 = 180° - \beta_1 - \theta_{1a}, \beta_5 + \alpha_1 = 180° - \beta_6 - \beta_2$$
$$\gamma_3 = \begin{cases} \tan^{-1}\left(\frac{d'}{L_2 - L_3}\right), & L_3 < L_2 \\ 90^o + \tan^{-1}\left(\frac{L_3 - L_2}{d'}\right), & L_3 \geq L_2 \end{cases}$$
$$b_1 = \sqrt{(b_2)^2 + (b)^2 - (2\ b\ b_2 \cos(\gamma_3))}$$
$$b_2 = \sqrt{(d')^2 + (L_2 - L_3)^2}$$

$\gamma_1 = 180° - \gamma_2 - \gamma_3$, $\gamma_2 = sen^{-1}\left(\frac{b}{b_1} sen(\gamma_3)\right)$, $\gamma_6 = \gamma_2 + \gamma_1 - \theta_{1b}$, $\alpha_1 = \gamma_6 + \alpha_{int} - \gamma_2$, $\alpha_2 + \gamma_5 = 180° - \gamma_6 - \alpha_{int}$.

3) The distances are obtained as:

left camera	right camera
$e_1 = c\frac{sen(\alpha_2)}{sen(\beta_2)}$	$f_2 = c\frac{sen(\alpha_1)}{sen(\gamma_2)}$
$e_2 = c\frac{sen(\beta_5 + \alpha_1)}{sen(\beta_2)}$	$f_1 = c\frac{sen(\alpha_2 + \gamma_5)}{sen(\gamma_2)}$

(9)

where L_1 is distance between the left border of the image and visual mark c_1 in left camera, L_3 is distance between the left border of the image and visual mark c_1 in right camera, L_2 is half of the width of the images captured by the cameras (320 pixels), c is distance between the visual marks c_1 and c_2 (42.7126 cm), α_{int} is angle between the c-line and the x-axis of the robot (4.1620°).

After the calibration of the Ready position, the robot is free to move, consequently the centroid coordinates (u_l, v_l) and (u_r, v_r) of the visual mark c_2 are unknown, therefore the distance c is also unknown (see Fig.4). In this part our main goal is obtain the position (coordinates (x, y, z)) of the visual mark c_2 (end-effector). To obtain the value of *z-axis*, we assume that both cameras are mounted at the same height and distance to the robot to ensure this condition, (we use an additional program to move the cameras until this condition is fulfilled) we establish the next relation

$$z = k_f \cdot u_l = k_f \cdot u_r \tag{10}$$

To obtain the coordinates we use an analog geometric model as shown in Fig.5, the difference now is we require to calculate the distance c' and the angle α' as shown in Fig.4. Using the same idea we obtain the coordinates

$$x = c' \cos(\alpha'), \quad y = c' \sin(\alpha')$$

Remark 1: Compared with the normal 3D model (8), the advantages of our model are: **General use,** using just three visual marks in the scene we can obtain the position of an unknown point. **Distance between the cameras and the scene**, it is possible to change the distance between the cameras and the robot while the experiment is carried on, our stereo model can recalculate them automatically as in (8). **Useful for applications**, our new stereo model does not require a fixed distance between the two cameras or a vergence angle ψ as in (8). The calculation equations are very simple, it can be used in different kinds of applications in industry.

III. Fuzzy visual servoing

The common use of this kind of industrial robot manipulator is first to save the trajectory of the end-effector in the articular controller, then the robot can move by the interior PID controller. The fuzzy visual servoing proposed in this section can give this robot more flexibility, because the "eye" of robot can see an unlimited number of points and send them to the articular controller automatically. In Section II, we proposed the vision algorithm and stereo vision model, which can generate a 3D position of robot's end-effector. In this section, we will use fuzzy control technique to design a fuzzy controller to move the industrial robot to a desired position. Since this fuzzy control is based on the visual information, we called it fuzzy visual servoing.

In order to simplify the experiment, we only control the three degrees of the 6 DOF robot, θ_1, θ_2 and θ_3 in Fig.4. We design three fuzzy controllers for each angle. The fuzzy controller will generate an angular change command and send it to the articular controller by the special instruction, named RAPL-II.

Selection of state and control variables.

Since the robot A465 and its lower level controller C550C require angular command, we define the joint angular error e and its derivation Δe as state variable

$$\begin{aligned} e &= \theta_i^{actual} - \theta_i^* \\ \Delta e &= e_i^{actual} - e_i^{anterior} \end{aligned} \tag{11}$$

where $i = 1, 2, 3$, θ_i^{actual} is the actual value of the *i-angle*, θ_i^* is the target value of the *i-angle,* e_i^{actual} is the actual value of the *i-error,* $e_i^{anterior}$ is the previous value of the *i-error*. These variables are inputs to each fuzzy module, the output (control variable) is also angular value. Since the visual information is position of the end-effector, we use the robot kinematics to transform the position information to angular. Now we can calculate the three angles by inverse kinematics model of the robot from the world frame (x, y, z), which is obtained from v_1, v_2 and v_3. We use pseudocode method [1] to calculate inverse kinematics. The pseudocode is as follows:

1) Calculate $\theta_1 = \begin{cases} 0 & y = 0 \\ \tan^{-1}\left(\frac{x}{y}\right) & y \neq 0 \end{cases}$

2) Calculate t_1, t_2, β_1, β_2 and β_4

$t_1 = \frac{L_1^2+r^2-L_2^2}{2L_1 r}, \quad t_2 = \frac{L_2^2+r^2-L_1^2}{2L_2 r}$

$\beta_1 = \cos^{-1}(t_1), \quad \beta_2 = \tan^{-1}\left(-\frac{x}{z}\right), \quad \beta_3 = \cos^{-1}(t_2)$

where $z = z - L_C$, $r = \sqrt{x^2+y^2}$

3) Calculate $\theta_2 = \beta_1 + \beta_2$, $\theta_3 = \beta_2 + \beta_3$

B. Creation of the fuzzy rules

The fuzzy sets are shown in Fig.7, where 1) Error (e)

fuzzy label	statement
LN1	Large Negative value
N1	Negative value
Z1	Zero value
P1	Positive value
LP1	Large Positive value

2) Derivation of the error (Δe)

fuzzy label	statement
LN2	Large Negative value
N2	Negative value
Z2	Zero value
P2	Positive value
LP2	Large Positive value

3) Output variable, angular value γ

fuzzy label	statement
LN3	Large Negative value
N3	Negative value
Z3	Zero value
P3	Positive value
LP3	Large Positive value

Although the range of the control variable is very difficult to adjust, we can construct our fuzzy rules by the robotics prior information. The corresponding fuzzy rules are obtained, the normal form as [14] is shown in Fig.8.

C. Implementation of fuzzy visual servoing and experimental results

The fuzzy visual servoing is applied in following steps: 1) The user set up coordinates for the end-effector of the robot and the target. 2) The stereo vision platform obtains visual information of the end-effector and the target. 3) Using the inverse kinematics model of the robot, we transform the positions of the end-effector and the target into their corresponding angles. 4) Apply our fuzzy visual servoing to move the robot to the target. 5) If the input variable $|e|$ is in desired accuracy for all fuzzy modules, the procedure is stopped, otherwise go to step 2. The complete control time for fuzzy visual servoing is about $15HZ$, which includes $10KHZ$ articular controller, $20HZ$ vision system and $1KHZ$ fuzzy controller. The real-time experimental results are shown in Fig.9.

We check 6 different ranges for the fuzzy output variable (γ) in this experiment. They are: 1) We set the

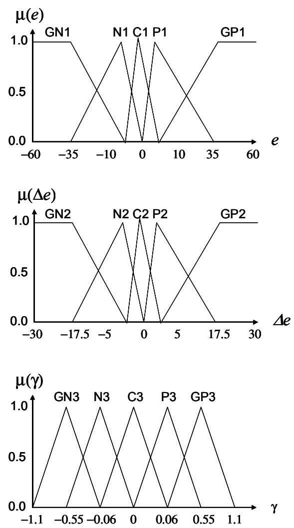


Fig. 7. Fuzzy sets for the variables e, Δe and γ with its final tuning values.

	LN2	N2	Z2	P2	LP2
LN1	LP3	LP3	LP3	Z3	Z3
N1	LP3	P3	P3	P3	Z3
Z1	LP3	P3	Z3	N3	LN3
P1	P3	Z3	N3	N3	LN3
LP1	Z3	Z3	LN3	LN3	LN3

Fig. 8. Fuzzy matrix filled with IF-THEN rules.

value range in $[-1.05, -0.7, -0.07, 0, 0.07, 0.7, 1.05]$ producing instability, it is shown in Fig.9.a. 2) We use the value range in $[-0.5, -0.25, -0.1, 0, 0.1, 0.25, 0.5]$, the end-effector cannot reach the desired object , it is shown in Fig. 9.b. 3) We select the value range in $[-0.54, -0.27, -0.07, 0, 0.07, 0.27, 0.54]$, the robot can reach the target in 23 seconds, however some unwanted oscillations appear, it is shown in Fig.9.c. 4) We use the value range in $[-0.2, -0.1, -0.05, 0, 0.05, 0.1, 0.2]$, the robot can reach the target in 23 seconds, the the fuzzy controller generates a smooth trajectory, but it is slow, it is shown in Fig.9.d. 5) We purpose the value range in $[-0.8, -0.4, -0.07, 0, 0.07, 0.4, 0.8]$, the robot can reach the target in 18 seconds, nevertheless there exist some oscillations, it is shown in Fig.9.e. 6) We choose the value range in $[-1.1, -0.55, -0.06, 0, 0.06, 0.55, 1.1]$, the robot can reach the final position in 6 seconds without oscillations.

IV. Conclusion

The main contributions of this paper are, 1) we developed and implement a new optical stereo vision model, 2) we successfully applied fuzzy visual servoing controller to control an industrial robot. Our stereo model can also be extended to other applications such as, mobile robots, 3D reconstruction, etc.

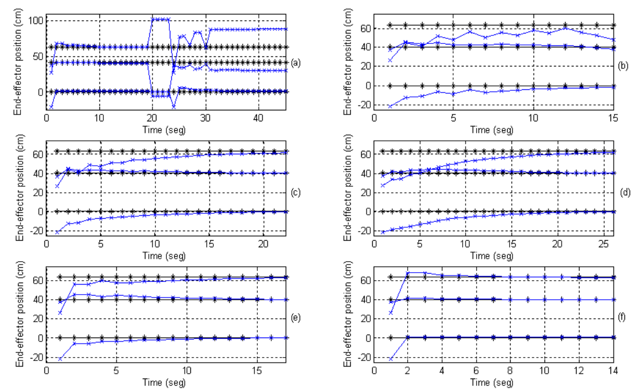


Fig. 9. Exprimental Results to used fuzzy visual servoing in the robot.

References

[1] T.H. Cormen, *Introduction to Algorithms*, 2nd Edition, MIT Press, 2001.
[2] J. Denavit and R.S. Hartenberg, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, *Journal of Applied Mechanics*, 1955, 215-221.
[3] B.Espiau, F.Chaumette and O.Rives, A new approach to visual servoing in robotics, *IEEE Trans. on Robotics and Automation*, Vol.8, No.3, 313-326, 1992
[4] E. Grosso, G. Metta, A. Oddera and G. Sandini, Robust visual servoing in 3-D reaching tasks, *IEEE Trans. on Robotics and Automation*, Vol.12, No.5, 732-741, 1996.
[5] K.Hashimoto and H.Kimura, Visual Servoing with Nonlinear Observer, *Proc. IEEE Int. Conf. Robotics and Automation*, 484-498, 1995.
[6] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision II*, Addison-Wesley Pub. Co., 1993.
[7] S.Hutchison, G.D.Hager and P.Corke, A tutorial on visual servo control, *IEEE Trans. on Robotics and Automation*, Vol.12, No.5, 651-670, 1996.
[8] R.Kelly, Robust asymptotically stable visual servoing of planar robots, *IEEE Trans. on Robotics and Automation*, Vol.12, No.5, 759-766, 1996.
[9] J.E.W. Mayhew and J.P Frisby, *3D Model Recognition from Stereoscopic Cues*, MIT Press, 1991.
[10] S.M Seitz and C.R Dyer, Complete Scene Structure from Four Point Correspondences, *Proc. 5th Int. Conf on Computer Vision*, Cambridge MA, 330-337, 1995.
[11] R. Szeliski, From Image to Model (and beyond): A Personal Restrospective, *Vision Interface 97*, Kelowna, B.C, May 21, 1997.
[12] C.J Taylor and D.J. Kriegman, Structure and Motion from Line Segments in Multiple Images, *IEEE Trans. on Pattern Analysis,* Machine Intelligence, 17(11), 1995.
[13] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 26-28, 1998.
[14] L.X.Wang, *Adaptive Fuzzy Systems and Control*, Englewood Cliffs NJ: Prentice-Hall, 1994.